

js_bokeh

November 26, 2021

1 bokeh

bokeh is one of the most mature and complete libraries using javascript.
[documentation](#) [source](#) [installation](#) [tutorial](#) [gallery](#)

```
[1]: from jupyter_bokeh import add_notebook_menu
      add_notebook_menu()
```

```
[1]: <IPython.core.display.HTML object>
```

1.1 example

```
[2]: from bokeh.io import output_notebook, output_file
      # The first line is only needed because bokeh creates a file <python>/Lib/run/runpy.
      #      ↪html
      # during the unit test. This line overrides these parameters. This should not be
      # needed otherwise.
      output_file("bokeh_temp.html")
      output_notebook()
```

```
[3]: import numpy as np

      from bokeh.plotting import figure, show, output_file
      from bokeh.models import HoverTool, ColumnDataSource
      from bokeh.sampledata.les_mis import data

      nodes = data['nodes']
      names = [node['name'] for node in sorted(data['nodes'], key=lambda x: x['group'])]

      N = len(nodes)
      counts = np.zeros((N, N))
      for link in data['links']:
          counts[link['source'], link['target']] = link['value']
          counts[link['target'], link['source']] = link['value']

      colormap = ["#444444", "#a6cee3", "#1f78b4", "#b2df8a", "#33a02c", "#fb9a99",
                  "#e31a1c", "#fdbf6f", "#ff7f00", "#cab2d6", "#6a3d9a"]

      xname = []
      yname = []
      color = []
```

```

alpha = []
for i, node1 in enumerate(nodes):
    for j, node2 in enumerate(nodes):
        xname.append(node1['name'])
        yname.append(node2['name'])

        alpha.append(min(counts[i,j]/4.0, 0.9) + 0.1)

        if node1['group'] == node2['group']:
            color.append(colormap[node1['group']])
        else:
            color.append('lightgrey')

source = ColumnDataSource(data=dict(
    xname=xname,
    yname=yname,
    colors=color,
    alphas=alpha,
    count=counts.flatten(),
))

p = figure(title="Les Mis Occurrences",
           x_axis_location="above",
           tools="hover,save,box_zoom,box_select,crosshair,reset",
           x_range=list(reversed(names)), y_range=names)

p.plot_width = 800
p.plot_height = 800
p.grid.grid_line_color = None
p.axis.axis_line_color = None
p.axis.major_tick_line_color = None
p.axis.major_label_text_font_size = "5pt"
p.axis.major_label_standoff = 0
p.xaxis.major_label_orientation = np.pi/3

p.rect('xname', 'yname', 0.9, 0.9, source=source,
       color='colors', alpha='alphas', line_color=None)

p.select_one(HoverTool).tooltips = [
    ('names', '@yname, @xname'),
    ('count', '@count'),
]

show(p) # show the plot

```

1.2 interactions

Go to [interaction](#).

1.2.1 synchronized moves

```
[4]: from bokeh.plotting import figure, gridplot, output_file, show

x = list(range(11))
y0 = x
y1 = [10-xx for xx in x]
y2 = [abs(xx-5) for xx in x]

# create a new plot
s1 = figure(width=250, plot_height=250, title=None)
s1.circle(x, y0, size=10, color="navy", alpha=0.5)

# create a new plot and share both ranges
s2 = figure(width=250, height=250, x_range=s1.x_range, y_range=s1.y_range, title=None)
s2.triangle(x, y1, size=10, color="firebrick", alpha=0.5)

# create a new plot and share only one range
s3 = figure(width=250, height=250, x_range=s1.x_range, title=None)
s3.square(x, y2, size=10, color="olive", alpha=0.5)

p = gridplot([[s1, s2, s3]], toolbar_location=None)

# show the results
show(p)
```

1.2.2 tab panes

```
[5]: from bokeh.models.widgets import Panel, Tabs
from bokeh.io import output_file, show
from bokeh.plotting import figure

p1 = figure(plot_width=300, plot_height=300)
p1.circle([1, 2, 3, 4, 5], [6, 7, 2, 4, 5], size=20, color="navy", alpha=0.5)
tab1 = Panel(child=p1, title="circle")

p2 = figure(plot_width=300, plot_height=300)
p2.line([1, 2, 3, 4, 5], [6, 7, 2, 4, 5], line_width=3, color="navy", alpha=0.5)
tab2 = Panel(child=p2, title="line")

tabs = Tabs(tabs=[ tab1, tab2 ])

show(tabs)
```

1.2.3 tables

```
[6]: from datetime import date
from random import randint

from bokeh.models import ColumnDataSource
from bokeh.models.widgets import DataTable, DateFormatter, TableColumn
from bokeh.io import output_file, show
```

```
data = dict(
    dates=[date(2014, 3, i+1) for i in range(10)],
    downloads=[randint(0, 100) for i in range(10)],
)
source = ColumnDataSource(data)

columns = [
    TableColumn(field="dates", title="Date", formatter=DateFormatter()),
    TableColumn(field="downloads", title="Downloads"),
]
data_table = DataTable(source=source, columns=columns, width=400, height=280)

show(data_table)
```

[7]: